Comparison of Morphological Pyramids for Multiresolution MIP Volume Rendering

Roerdink, Jos B.T.M.

Published in:
EPRINTS-BOOK-TITLE

Publication date:
2002

Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2002)
D. Ebert, P. Brunet, I. Navazo (Editors)

Comparison of Morphological Pyramids for Multiresolution MIP Volume Rendering

Jos B.T.M. Roerdink

Institute for Mathematics and Computing Science
University of Groningen
P.O. Box 800, 9700 AV Groningen, The Netherlands
Email: roe@cs.rug.nl

Abstract

We recently proposed a multiresolution representation for maximum intensity projection (MIP) volume rendering based on morphological adjunction pyramids which allow progressive refinement and have the property of perfect reconstruction. In this algorithm the pyramidal analysis and synthesis operators are composed of morphological erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis. Here we introduce an alternative pyramid scheme in which a morphological opening instead of an erosion is used for pyramidal analysis. As a result, the approximation accuracy when rendering from higher levels of the pyramid is improved.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Interaction techniques. I.4.10 [Image processing and Computer vision]: Image Representation, Hierarchical, Morphological.

1. Introduction

This paper is concerned with multiresolution algorithms for Maximum Intensity Projection (MIP) volume rendering, where one computes the maximum value along lines through a 3-D data set. This algorithm is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data, both because of its computational simplicity, and also since this method does not change the original data values, since only maxima along viewing rays are computed.

When using interactive rendering of volume data sets, data size presents a major problem. One solution to deal with this is to introduce multiresolution models, which can be used to visualize data in preview mode. As long as a user is interacting with the data, only a coarse version of the data is used, thus lowering rendering time. When interaction ceases, details of the data are successively taken into account ('progressive refinement'). The purpose of using multiresolution models in this case is therefore improved user interaction, since the response time of the system drops. The price to pay is of course image quality, i.e. less detailed renderings, in preview mode. Whether such loss of detail is acceptable is a question which has to be answered in each specific application.

In a previous paper [12] (see [11] for an expanded version) we proposed a pyramid scheme for MIP volume rendering with progressive refinement based on morphological pyramids [3, 6]. Such pyramids, which involve nonlinear spatial filtering by morphological operators [5, 14], systematically split the volume data into approximation and detail signals. That is, as the level of the pyramid is increased, spatial features of increasing size are extracted. Even though the morphological operators are nonlinear and non-invertible, the pyramid scheme does allow perfect reconstruction as well as progressive refinement. After the pyramid has been constructed the original volume data can be discarded, because of the perfect reconstruction property. Also, only integer computations are required.

Morphological pyramids are appropriate in the context of MIP because the morphological operations of erosion and dilation (involving the computation of minima and maxima of voxel values in a local neighbourhood) are compatible with the maximum computation involved in MIP, just as linear pyramid [1] or wavelet [4, 7, 8, 10, 17, 18] represen-

tations are the right tool for the case of linear X-ray rendering. Also, the feature extraction capabilities of morphological operators are automatically incorporated within the volume rendering process. This allows the extraction of features based on spatial information. In this way, morphological pyramids combine feature extraction with accelerated rendering in preview mode.

In the algorithm as proposed in [12] the pyramidal analysis and synthesis operators are composed of morphological erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis. Such pyramids, in which the analysis and synthesis involve an erosion/dilation pair, are called *adjunction* pyramids [3]. One of the problems with this type of pyramid is that too few small features present in the data are retained in higher levels of the pyramid. To put it differently, the detail signals are 'too large'. To improve the effectiveness of feature extraction we introduce here an alternative pyramid scheme in which a morphological opening instead of an erosion is used for pyramidal analysis (the pyramidal synthesis operator is still a dilation). This pyramid has been studied in morphological image processing by Sun and Maragos [15], see also [3], and therefore is referred to as the *Sun-Maragos pyramid* below. In this paper, we only use so-called *flat* pyramids, where minima and maxima are computed in a local neighbourhood of each voxel, so that no new grey values are introduced in the analysis of the volume data.

In order to allow for compression domain rendering, it is essential to use a fast MIP implementation which can work directly on the data structures used to represent the pyramid. Similar issues have been studied in the case of linear (Laplacian) pyramids [2]. In the examples below we will use a voxel projection method with an efficient volume data storage scheme, by histogram-based sorting of 'interesting' voxels according to grey value, and storing these in a value-sorted array of voxel positions; an additional array contains the cumulative histogram values [9]. To maintain computational efficiency, a modification of the structure of the progressive rendering algorithm used for adjunction pyramids is necessary, which will be described below.

The remainder of this paper is organized as follows. Section 2 recalls a few preliminaries on morphological operators, and gives the definition of adjunction and Sun-Maragos pyramids. In section 3 we first recall the MMIP algorithm based on adjunction pyramids, and then introduce the new MMIP algorithm based on Sun-Maragos pyramids. A comparison of the two types of pyramid is presented in section 4. Section 5 contains a summary and discussion of future work.

2. Multiresolution volume rendering by morphological pyramids

In this section we first define some elementary morphological operators [5, 14]. Next we summarize the basics of (non)linear pyramids. Then we introduce morphological pyramids, in particular adjunction and Sun-Maragos pyramids.

2.1. Morphological operators

Let f be a signal with domain $F \subseteq \mathbb{Z}^d$, and A a subset of $\mathbb{Z}^d$ called the *structuring element*. The *dilation* $\delta_A(f)$ and *erosion* $\varepsilon_A(f)$ of f by A are defined by

$$\delta_A(f)(x) = \max_{y \in A, x-y \in F} f(x-y), \tag{1}$$

$$\varepsilon_A(f)(x) = \min_{y \in A, x+y \in F} f(x+y). \tag{2}$$

Dilation and erosion simply replace each signal value by the maximum or minimum in a neighbourhood defined by the structuring element A. The *opening* $\alpha_A(f)$ and *closing* $\phi_A(f)$ of f by A are defined by

$$\alpha_A(f)(x) = \delta_A(\varepsilon_A(f))(x),$$
$$\phi_A(f)(x) = \varepsilon_A(\delta_A(f))(x).$$

So openings and closings are products of a dilation and an erosion. The opening has the property that it is increasing ($f \leq g$ implies that $\alpha_A(f) \leq \alpha_A(g)$), anti-extensive ($\alpha_A(f) \leq f$) and idempotent ($\alpha_A(\alpha_A(f)) = \alpha_A(f)$). Similar properties hold for the closing, with the difference that closing is extensive ($\phi_A(f) \geq f$). The opening eliminates signal peaks, the closing valleys.

2.2. Pyramids

Consider signals in a d-dimensional signal space V_0, which is assumed to be the set of functions on (a subset of) the discrete grid $\mathbb{Z}^d$, where $d = 2$ or $d = 3$ (image and volume data), that take values in a finite set of nonnegative integers.

The general structure of linear as well as nonlinear pyramids is as follows. From an initial data set f_0, approximations $\{f_j\}$ of increasingly reduced size are computed by a reduction operation:

$$f_j = \text{REDUCE}(f_{j-1}), \qquad j = 1, 2, \ldots L.$$

Here j is called the level of the decomposition. The set $\{f_0, f_1, \ldots, f_L\}$ is referred to as an *approximation pyramid*. In the case of a Gaussian pyramid, the REDUCE operation consists of Gaussian low-pass filtering followed by downsampling [1]. An approximation error associated to f_{j+1} may be defined by taking the difference between f_j and an expanded version of f_{j+1}:

$$d_j = f_j \dot{-} \text{EXPAND}(f_{j+1}). \tag{3}$$

The set $d_0, d_1, \ldots, d_{L-1}, f_L$ is referred to as a *detail pyramid*. Here $\dot{-}$ is a generalized subtraction operator. Assuming there exists an associated generalized addition operator $\dot{+}$ such that, for all j,

$$\hat{f}_j \dot{+} (f_j \dot{-} \hat{f}_j) = f_j, \quad \text{where } \hat{f}_j = \text{EXPAND}(\text{REDUCE}(f_j)),$$

we have *perfect reconstruction*, that is, f_0 can be exactly reconstructed by the recursion

$$f_j = \text{EXPAND}(f_{j+1}) \dot{+} d_j, \quad j = L-1, \ldots, 0. \tag{4}$$

For the linear case, the detail pyramid is called a Laplacian pyramid, $\dot{+}$ and $\dot{-}$ are ordinary addition and subtraction, and the EXPAND operation consists of upsampling followed by Gaussian low-pass filtering [1].

To guarantee that information lost during analysis can be recovered in the synthesis phase in a non-redundant way, one needs the so-called *pyramid condition*:

$$\text{REDUCE}(\text{EXPAND}(f)) = f \text{ for all } f. \tag{5}$$

2.3. Morphological pyramids

In the case of morphological pyramids, the REDUCE and EXPAND operations involve morphological filtering instead of linear Gaussian filtering [3, 6]. We first review the case of adjunction pyramids as used in [12] for MIP rendering, and then define the Sun-Maragos pyramid.

2.3.1. Adjunction pyramid

Morphological adjunction pyramids [3] involve the morphological operators of dilation $\delta_A(f)$ and erosion $\varepsilon_A(f)$ with structuring element A defined in (1) and (2), respectively. In this case the REDUCE and EXPAND operators are denoted by $\psi_A^{\uparrow}$ and $\psi_A^{\downarrow}$, respectively, and have the form

$$\text{REDUCE} : \psi_A^{\uparrow}(f) = \text{DOWNSAMPLE}(\varepsilon_A(f)), \tag{6}$$

$$\text{EXPAND} : \psi_A^{\downarrow}(f) = \delta_A(\text{UPSAMPLE}(f)), \tag{7}$$

where the arrows indicate transformations to higher (coarser) or lower (finer) levels of the pyramid. Here DOWNSAMPLE and UPSAMPLE denote downsampling and upsampling by a factor of 2 in each spatial dimension. The pyramid condition (5) is satisfied, if there exists an $a \in A$ such that the translates of a over an even number of grid steps are never contained in the structuring element A; see [3] for more details.

The generalized addition and subtraction operators $\dot{+}$ and $\dot{-}$ appearing in the definition (3) of the detail signals and the reconstruction equation (4) may be taken as ordinary addition and subtraction. However, the following alternative definition is possible. In an adjunction pyramid, the product $\psi_A^{\downarrow}\psi_A^{\uparrow}$ is an *opening*, i.e. an operator which is increasing, anti-extensive and idempotent. The anti-extensivity property means that $\psi_A^{\downarrow}\psi_A^{\uparrow}(f) \leq f$. Therefore, we can define the generalized addition and subtraction operators by (cf. [3]):

$$t \dot{+} s = t \vee s = \max(t,s), \qquad t \dot{-} s = \begin{cases} t, & \text{if } t > s \\ 0, & \text{if } t = s \end{cases} \tag{8}$$

where 0 is the smallest image or voxel value possible. As a consequence, the detail signals are nonnegative:

$$d_j(n) = f_j(n) \dot{-} \psi_A^{\downarrow}(f_{j+1})(n) = f_j(n) \dot{-} \psi_A^{\downarrow}\psi_A^{\uparrow}(f_j)(n) \geq 0. \tag{9}$$

Note that the definition of $\dot{-}$ in (8) implies that the detail signal $d_j(n)$ equals $f_j(n)$, except at points n for which $f_j(n) = \psi_A^{\downarrow}\psi_A^{\uparrow}(f_j)(n)$, where $d_j(n) = 0$. So, detail signals are not 'small' in regions where the structuring element does not fit well to the data.

For an adjunction pyramid with the generalized addition being defined as the maximum operation (see (8)), the reconstruction takes a special form. Making use of the fact that $\psi_A^{\downarrow}$ is a dilation, hence commutes with the maximum operation, we derive from (4) and (8):

$$f = \psi_A^{\downarrow L}(f_L) \vee \bigvee_{k=0}^{L-1} \psi_A^{\downarrow k}(d_k), \tag{10}$$

where L is the decomposition depth, $\psi_A^{\downarrow k}$ denotes k-fold composition of $\psi_A^{\downarrow}$ with itself, and $\bigvee_{k=0}^{N} g_k$ is shorthand notation for the maximum of the functions g_k, i.e. its value at a point x equals $g_0(x) \vee g_1(x) \vee \ldots \vee g_N(x) = \max(g_0(x), g_1(x), \ldots, g_N(x))$. This representation is quite similar to the (linear) Laplacian pyramid representation [1]. The main difference is that sums have been replaced by maxima.

2.3.2. Sun-Maragos pyramid

Now the analysis operator is an opening α_A. In this case the REDUCE and EXPAND operators are given by:

$$\text{REDUCE} : \psi_A^{\uparrow}(f) = \text{DOWNSAMPLE}(\alpha_A(f)), \tag{11}$$

$$\text{EXPAND} : \psi_A^{\downarrow}(f) = \delta_A(\text{UPSAMPLE}(f)). \tag{12}$$

Note that the EXPAND operator is identical to that of the adjunction pyramid, cf. (7). In this case we cannot use (8), so we define $\dot{+}$ and $\dot{-}$ as ordinary addition and subtraction. The pyramid condition (5) is satisfied if the structuring element A contains the origin of $\mathbb{Z}^d$ and the translates of the origin over an even number of grid steps are never contained in A; see [3].

3. Multiresolution MIP algorithms

The basic idea of multiresolution MIP is to use a pyramid representation in which it is possible to interchange the MIP operator (computing maxima along the line of sight) with the pyramidal synthesis operator. Then the MIP operation can be performed on a coarse level, where the size of the data is reduced, before performing a 2-D EXPAND operation to a finer level, thus leading to a computationally efficient algorithm. For the pyramids defined above, such commutativity of MIP and pyramid synthesis holds because both the upsampling operation and the dilation δ_A commute with the maximum operation [11, 12].

In the following, the MIP operation is denoted by $\mathcal{M}_\Theta$, where the viewing coordinates are denoted by a vector $\Theta = (\theta, \phi, \alpha)$. Here θ and ϕ are two angles defining the projection direction vector which is perpendicular to the view plane, and the angle α gives the orientation of the view plane with respect to this projection direction. Successive approximations of the MIP of f are denoted by $\hat{\mathbf{M}}_\Theta^{(j)}(f)$, $j = L, L-1, \ldots, 0$. These approximations all have the same size in the image plane, because they are reconstructed from higher levels in the pyramid to the size of the MIP of f. We now summarize the progressive refinement algorithms for the different pyramids.

Algorithm 3.1 Multiresolution MIP algorithm based on detail pyramid. Synthesis operator must be a dilation.

1: INPUT: detail pyramid sequence d_0, d_1, …, d_{L-1}, f_L of input data set f.
2: OUTPUT: progressively refined approximation images.
3:
4: Choose an orientation Θ of the viewing coordinate system.
5: $\hat{\mathbf{M}}_\Theta^{(L)}(f) = \psi_{\tilde{A}}^{\downarrow L}(\mathcal{M}_\Theta(f_L))$ (∗Coarsest approximation∗)
6:
7: **for** $j = L$ **to** 1 **do** (∗Progressively refine∗)
8: $f_{j-1} = \psi_A^{\downarrow}(f_j) \dot{+} d_{j-1}$
9: $\hat{\mathbf{M}}_\Theta^{(j-1)}(f) = \psi_{\tilde{A}}^{\downarrow j-1}(\mathcal{M}_\Theta(f_{j-1}))$
10: **end for**

Algorithm 3.2 Multiresolution MIP algorithm based on an adjunction pyramid.

1: INPUT: adjunction detail pyramid sequence d_0, d_1, …, d_{L-1}, f_L of input data set f.
2: OUTPUT: progressively refined approximation images.
3:
4: Choose an orientation Θ of the viewing coordinate system.
5: $\hat{\mathbf{M}}_\Theta^{(L)}(f) = \psi_{\tilde{A}}^{\downarrow L}(\mathcal{M}_\Theta(f_L))$ (∗Coarsest approximation∗)
6:
7: **for** $j = L$ **to** 1 **do** (∗Progressively refine∗)
8: $\hat{\mathbf{M}}_\Theta^{(j-1)}(f) = \psi_{\tilde{A}}^{\downarrow j-1}(\mathcal{M}_\Theta(d_{j-1})) \vee \hat{\mathbf{M}}_\Theta^{(j)}(f)$.
9: **end for**

Algorithm 3.3 Multiresolution MIP algorithm based on approximation pyramid. Synthesis operator must be a dilation.

1: INPUT: approximation pyramid sequence f_0, f_1, …, f_{L-1}, f_L of input data set f.
2: OUTPUT: progressively refined approximation images.
3:
4: Choose an orientation Θ of the viewing coordinate system.
5: **for** $j = L$ **to** 0 **do** (∗Progressively refine∗)
6: $\hat{\mathbf{M}}_\Theta^{(j)}(f) = \psi_{\tilde{A}}^{\downarrow j}(\mathcal{M}_\Theta(f_j))$.
7: **end for**

3.1. Refinement from a detail pyramid

In the general case, when performing progressive refinement one must first do a one-level 3-D reconstruction of f_{j-1} from f_j, see (4). Then the MIP operator can be applied to the volume data, and the resulting 2-D image is finally expanded $j-1$ times by a 2-D EXPAND operator which has the same form as (7), that is, 2-D upsampling followed by a 2-D dilation, but with a structuring element $\tilde{A}$ which is the MIP of A [11,12]. This algorithm is applicable for all detail pyramids where the synthesis operator is a dilation. The steps are as follows (please refer to the pseudocode in Algorithm 3.1). In line 5 a coarse approximation is computed by first letting the MIP operator $\mathcal{M}_\Theta$ act on the level-L approximation data f_L, and then L times applying the 2-D expand operator $\psi_{\tilde{A}}^{\downarrow}$. Then follows the progressive refinement: in line 8 the level-j data is expanded one level downwards by $\psi_A^{\downarrow}$ and the result 'added' to the detail signal d_{j-1} on level $j-1$, yielding an approximation f_{j-1}; in line 9 this approximation is projected by $\mathcal{M}_\Theta$, and finally the 2-D expand operator $\psi_{\tilde{A}}^{\downarrow}$ is applied $j-1$ times.

3.2. Adjunction pyramid

For an adjunction pyramid where detail signals and reconstructions are computed by using ordinary addition and subtraction, Algorithm 3.1 is applicable. However, when using (8) we can refine directly in the image plane, as follows from (10). This case has been treated in [11,12]. The pseudo-code of this algorithm is given in Algorithm 3.2.

The structure of the algorithm is as follows. From a level-j approximation, the next approximation on level $j-1$ is obtained by first computing the MIP of d_{j-1}, then $j-1$ times applying the 2-D pyramid synthesis operator $\psi_{\tilde{A}}^{\downarrow}$ to the projection, and finally taking the maximum of the image so obtained with the previous approximation. It is clear that $\hat{\mathbf{M}}_\Theta^{(j-1)}(f) \geq \hat{\mathbf{M}}_\Theta^{(j)}(f)$, since from (9) the details signals d_{j-1} are nonnegative. So the projections increase pointwise as one goes down the pyramid. This algorithm is very similar to that of wavelet splatting [7,8,16]. The main differences are that (i) linear summation of voxel values is replaced by maximum computation, and (ii) linear wavelet filters are replaced by morphological filters.

3.3. Sun-Maragos pyramid

When performing progressive refinement with a Sun-Maragos pyramid, Algorithm 3.1 is in principle applicable. But here a difficulty arises. For efficiency reasons, one would like to store the data in a value-sorted array of voxel positions, where the 'non-interesting' voxels have been removed, see Section 3.4. In particular, zero voxels should be deleted, because usually these make up a very large fraction of the total data. The problem now is that during reconstruction, one must first do a one-level 3-D reconstruction of f_{j-1} from f_j

which involves a 3-D dilation (see line 8 in Algorithm 3.1), and there is no obvious way to efficiently compute dilations of data stored as value-sorted arrays. One solution would be to convert the data first to 3-D array format, compute the dilation, and then revert back to value-sorted array format. But this requires a considerable computation time, thus obviating the advantage of pyramids to give a quick preview of the volume data. Storing the data in 3-D array format is not a good alternative because data size and hence rendering time goes up dramatically because of the need to process large amounts of zero voxels.

So, we resort to the solution of performing progressive refinement from an approximation pyramid, instead of a detail pyramid. The corresponding algorithm is given in Algorithm 3.3. Here the MIP operator $\mathcal{M}_\Theta$ acts directly on the level-j approximation data f_j, $j = L, L-1, \ldots, 0$, followed by j-fold application of the 2-D expand operator $\psi_A^\downarrow$.

For the morphological pyramids used here, storing the approximation data f_0, f_1, ..., f_{L-1}, f_L does require memory storage of the same order of magnitude as storing the detail and coarsest approximation data d_0, d_1, ..., d_{L-1}, f_L (see Section 4).

3.4. Implementation

The MIP operations required in the MMIP algorithms can be implemented by a simple object order voxel projection method [9]. In this method, one loops through the volume, projects all voxels to the image plane with each voxel contributing to exactly one pixel, and accumulates values at pixel locations by maximum computation. The final result is independent of the order in which the voxels are visited. This method also uses an efficient volume data storage scheme, by histogram-based sorting of 'interesting' voxels according to grey value, and storing these in a value-sorted array of voxel positions. An additional array contains the cumulative histogram values. In the experiments to be discussed in Section 4, all levels of the pyramid were created and stored as value-sorted arrays. We define interesting voxels simply as those with a nonzero grey value (zero voxel values never contribute to pixel maxima). In practice, especially for angiographic data, a substantial reduction (sometimes more than 95%) in the amount of voxels to be processed is thus obtained. It is essential to observe that compression domain rendering is possible: the only operation required on the approximation and/or detail volume data is MIP, and this can be directly carried out on data in value-sorted array format. Moreover, since the voxels are available in sorted order, one may project voxels from low to high value, so that old values in the 2-D image array can simply be overwritten by current values.

For non-axial views, that is, projections where the axes of the viewing coordinate system are not parallel to the axes of the original grid of volume data, the above algorithm needs to be adapted. To prevent the formation of holes in the projection image due to undersampling, a continuous function has first to be reconstructed from the discrete data. For that purpose, morphological sampling can be used, an interpolation method well adapted to the nonlinear character of MIP. As shown in [11], the result of this analysis is that after voxel projection a final morphological closing of the projection images has to be applied.

4. Experiments

Experiments were carried out on a PC with a 1.9 GHz Pentium 4 processor and 512 Mb memory. We performed MMIP rendering using a 2-level adjunction and Sun-Maragos pyramid, respectively, with a $2 \times 2 \times 2$ structuring element. Three data sets were used, all of size 256^3: MRA angiography data (1.25% nonzero voxels), CT data of a human head (29.4% nonzero voxels), and CT data of a bonsai tree (16.5% nonzero voxels). The sampling distance in the view plane was taken equal to the sampling distance of the original volume data. Creation of the pyramid took about 5 seconds in all cases. To remove some uniform background noise, voxels with value below a small threshold (4% of the maximum grey value) were first set to zero. Rendering times were found to be almost independent of view angle. Sizes of approximation and detail data in value-sorted array format and rendering times (averages over 50 runs) of the successive levels of the pyramid are given in Tables 1-3. For comparison, the numbers for direct MIP rendering of the full-size volume data are given as well. All times are excluding I/O. The table also shows the relative L_2-error $\mathcal{E}^{(j)}$ between a level-j approximation image $\hat{\mathbf{M}}_\Theta^{(j)}$ and the full image $\hat{\mathbf{M}}_\Theta^{(0)}$:

$$\mathcal{E}^{(j)} = \|\mathbf{M}_\Theta(f) - \hat{\mathbf{M}}_\Theta^{(j)}(f)\|_2 / \|\mathbf{M}_\Theta(f)\|_2. \qquad (13)$$

The timings show that computing a level-2 or level-1 approximation takes considerably less time than a full-size MIP.

Figures 1-3 show successive approximations, for both the adjunction and Sun-Maragos pyramid. For enhanced display purposes, we show the images in reverse-video mode (high intensity corresponding to low grey value). From visual inspection of the figures, it is clear that in higher levels of the adjunction pyramid small details are quickly removed. In contrast, the Sun-Maragos pyramid preserves small details much better, as is also evident from the behaviour of the L_2-error in Tables 1-3. Note that this accuracy improvement is possible with rendering times and memory usage comparable to those of adjunction pyramids.

5. Discussion

We have discussed two types of morphological pyramid for multiresolution maximum intensity projection (MMIP) volume rendering with progressive refinement and perfect reconstruction, which can be used to visualize data in preview mode. Such pyramids combine the feature extraction

Table 1: *Data sizes (value-sorted array format) and rendering times of MIP (full image) and MMIP (progressive rendering) of MRA angiography data (cf. Fig. 1) using an adjunction pyramid (Algorithm 3.2) and a Sun-Maragos pyramid (Algorithm 3.3), respectively. Error denotes the relative L_2-error, as defined in (13), between approximation image and full image.*

Adjunction pyramid	rendered data	size (kbytes)	time (s)	error
level 2 approx.	f_2	0.85	0.02	0.96
add detail level 1	d_1	30.2	0.03	0.57
add detail level 0	d_0	801.6	0.26	0.0
full image	f_0	838.5	0.27	

Sun-Maragos pyramid	rendered data	size (kbytes)	time (s)	error
level 2 approx.	f_2	5.04	0.02	0.74
level 1 approx.	f_1	63.4	0.04	0.40
level 0 approx.	f_0	838.5	0.27	0.0
full image	f_0	838.5	0.27	

Table 2: *Same as Table 1, but for CT head data, see Fig. 2.*

Adjunction pyramid	rendered data	size (kbytes)	time (s)	error
level 2 approx.	f_2	277	0.13	0.39
add detail level 1	d_1	2052	0.70	0.15
add detail level 0	d_0	17231	5.86	0.0
full image	f_0	19715	6.64	

Sun-Maragos pyramid	rendered data	size (kbytes)	time (s)	error
level 2 approx.	f_2	316	0.12	0.22
level 1 approx.	f_1	2454	0.81	0.10
level 0 approx.	f_0	19715	6.32	0.0
full image	f_0	19715	6.64	

capabilities of morphological operators with the acceleration gained by rendering multiresolution data. Of course, the speed-up gained by rendering coarser approximations is counterbalanced by a loss in image quality. The morphological operators used in constructing these pyramids remove spatial details of size proportional to 2^j, where j is the level of the pyramid. However, in contrast to linear multiresolution approaches, such as those based on wavelets, no smoothing of data takes place. Typical use of such pyramids in in preview mode, where the user is continuously rotating the viewpoint. For such interactive display, some loss in image quality is generally acceptable. When interaction ceases, details of the data can then be successively taken into account to quicly generate a high resolution view.

Table 3: *Same as Table 1, but for CT data of a bonsai tree, see Fig. 3.*

Adjunction pyramid	rendered data	size (kbytes)	time (s)	error
level 2 approx.	f_2	89	0.05	0.44
add detail level 1	d_1	922	0.31	0.18
add detail level 0	d_0	9606	3.04	0.0
full image	f_0	11061	3.51	

Sun-Maragos pyramid	rendered data	size (kbytes)	time (s)	error
level 2 approx.	f_2	130	0.06	0.34
level 1 approx.	f_1	1245	0.42	0.20
level 0 approx.	f_0	11061	3.51	0.0
full image	f_0	11061	3.51	

A first MMIP algorithm, discussed in [11, 12], is based upon morphological adjunction pyramids, where the pyramidal analysis and synthesis operators are composed of morphological erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis. In this paper we introduced an alternative pyramid scheme, the so-called Sun-Maragos pyramid, in which an morphological opening instead of an erosion is used for pyramidal analysis. As a result, effectiveness of feature extraction in higher levels of the pyramid is substantially improved, as indicated by enhanced visual quality as well as smaller mean-squared errors. In order to maintain efficient rendering of the Sun-Maragos pyramid, we use an approximation pyramid instead of a detail pyramid. Experiments indicated that this accuracy improvement is possible with rendering times and memory usage comparable to those of the adjunction pyramid.

A property of the morphological operators (erosion, dilation, opening, closing) used in constructing these pyramids is that they remove spatial details without regard to the connectivity properties of the volume data. For example, in the case of angiography data, both small veins and small parts of larger veins are removed in higher levels of the pyramid. One could imagine situations where one would prefer that the morphological operations are shape-preserving, i.e. spatial structures should be either removed completely or retained completely. This would require the use of so-called connected operators [13], a topic we plan to study in future work.

(a) original $\hat{\mathbf{M}}_\Theta^{(0)}$

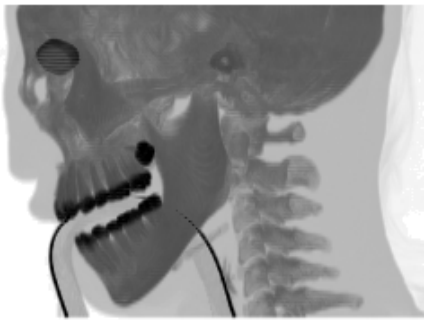
(b) level 1 approx. $\hat{\mathbf{M}}_\Theta^{(1)}$

(c) level 2 approx. $\hat{\mathbf{M}}_\Theta^{(2)}$

(d) level 1 approx. $\hat{\mathbf{M}}_\Theta^{(1)}$

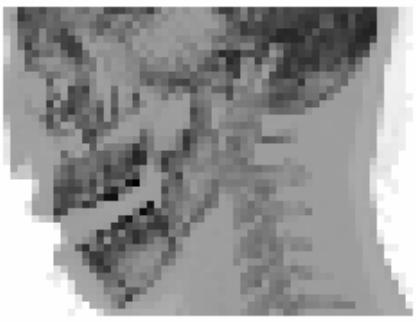
(e) level 2 approx. $\hat{\mathbf{M}}_\Theta^{(2)}$

Figure 1: *MMIP reconstruction of MR angiography data from a 2-level morphological pyramid using a* $2 \times 2 \times 2$ *structuring element. (a): original; (b) and (c): adjunction pyramid; (d) and (e): Sun-Maragos pyramid.*



(a) original $\hat{\mathbf{M}}_\Theta^{(0)}$

(b) level 1 approx. $\hat{\mathbf{M}}_\Theta^{(1)}$



(c) level 2 approx. $\hat{\mathbf{M}}_\Theta^{(2)}$

(d) level 1 approx. $\hat{\mathbf{M}}_\Theta^{(1)}$

(e) level 2 approx. $\hat{\mathbf{M}}_\Theta^{(2)}$

Figure 2: *MMIP reconstruction of CT head data (courtesy D. C. Hemmy, Medical College of Wisconsin) from a 2-level morphological pyramid using a* $2 \times 2 \times 2$ *structuring element. (a): original; (b) and (c): adjunction pyramid; (d) and (e): Sun-Maragos pyramid.*

(a) original $\hat{\mathbf{M}}_\Theta^{(0)}$

(b) level 1 approx. $\hat{\mathbf{M}}_\Theta^{(1)}$

(c) level 2 approx. $\hat{\mathbf{M}}_\Theta^{(2)}$

(d) level 1 approx. $\hat{\mathbf{M}}_\Theta^{(1)}$

(e) level 2 approx. $\hat{\mathbf{M}}_\Theta^{(2)}$

Figure 3: *MMIP reconstruction of CT data of a bonsai tree (courtesy S. Roettger, University of Stuttgart) from a 2-level morphological pyramid using a* $2 \times 2 \times 2$ *structuring element. (a): original; (b) and (c): adjunction pyramid; (d) and (e): Sun-Maragos pyramid.*

References

1. Burt, P. J., and Adelson, E. H. The Laplacian pyramid as a compact image code. *IEEE Trans. Communications 31* (1983), 532–540.

2. Ghavamnia, M. H., and Yang, X. D. Direct rendering of Laplacian pyramid compressed volume data. In *Proc. IEEE Visualization '95* (1995), IEEE Computer Society Press, pp. 192–199.

3. Goutsias, J., and Heijmans, H. J. A. M. Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids. *IEEE Trans. Image Processing 9*, 11 (2000), 1862–1876.

4. Grosso, R., and Ertl, T. Biorthogonal wavelet filters for frequency domain volume rendering. In *Proceedings of Visualization in Scientific Computing '95* (1995), J. van Wijk, R. Scateni, and P. Zanarini, Eds.

5. Heijmans, H. J. A. M. *Morphological Image Operators*, vol. 25 of *Advances in Electronics and Electron Physics, Supplement*. Academic Press, New York, 1994.

6. Heijmans, H. J. A. M., and Goutsias, J. Multiresolution signal decomposition schemes. Part 2: morphological wavelets. *IEEE Trans. Image Processing 9*, 11 (2000), 1897–1913.

7. Lippert, L., and Gross, M. H. Fast wavelet based volume rendering by accumulation of transparent texture maps. *Computer Graphics Forum 14*, 3 (1995), 431–443.

8. Lippert, L., Gross, M. H., and Kurmann, C. Compression domain volume rendering for distributed environments. In *Proc. Eurographics'97* (1997), pp. 95–107.

9. Mroz, L., König, A., and Gröller, E. Maximum intensity projection at warp speed. *Computers & Graphics 24* (2000), 343–352.

10. Muraki, S. Volume data and wavelet transforms. *IEEE Computer Graphics and Applications 13*, 4 (1993), 50–56.

11. Roerdink, J. B. T. M. Multiresolution maximum intensity volume rendering by morphological adjunction pyramids. Tech. Rep. 2001-9-03, Institute for Mathematics and Computing Science, University of Groningen, the Netherlands, July 2001. Under review.

12. Roerdink, J. B. T. M. Multiresolution maximum intensity volume rendering by morphological pyramids. In *Data Visualization 2001. Proc. Joint Eurographics – IEEE TCVG Symposium on Visualization, May 28-30, 2001, Ascona, Switzerland*, D. Ebert, J. M. Favre, and R. Peikert, Eds. Springer, Wien, New York, 2001, pp. 45–54.

13. Salembier, P., and Serra, J. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing 4*, 8 (1995), 1153–1160.

14. Serra, J. *Image Analysis and Mathematical Morphology*. Academic Press, New York, 1982.

15. Sun, F. K., and Maragos, P. Experiments on image compression using morphological piramids. In *SPIE Conf. Vis. Comm. Im. Proc. IV* (1989), vol. 1199, pp. 1303–1312.

16. Westenberg, M. A., and Roerdink, J. B. T. M. Frequency domain volume rendering by the wavelet X-ray transform. *IEEE Trans. Image Processing 9*, 7 (2000), 1249–1261.

17. Westenberg, M. A., and Roerdink, J. B. T. M. An extension of Fourier-wavelet volume rendering by view interpolation. *J. Math. Imag. Vision 14*, 2 (2001), 103–115.

18. Westermann, R., and Ertl, T. A multiscale approach to integrated volume segmentation and rendering. In *Proc. Eurographics'97, Vienna* (1997), D. Fellner and L. Szirmay-Kalos, Eds., vol. 16 (3), pp. C–117–C–127.